

JS基础

Day01

课上练习

4.2 自增自减练习

需求: 回答下面每道题的结果

```
// 需求: 回答下面代码打印的结果
// 题目1:
var g = 3;
var h = ++g + g++;
console.log(g, h);

// 题目2:
var a = 10;
var b = 5;
var c = ++a + ++b;
var d = c++ + ++c;
var e = ++d + d++;
console.log(a, b, c, d, e);
```

答案

g是5, h是8
a, b, c, d, e的值是 11 6 19 38 74

案例

2.4 变量复制

```
// 需求: 把变量a的值, 赋值给变量b
// 思路: var 变量b, 然后用 =(赋值运算符), 把右侧变量a赋值给变量b

var a = 10; // 以后写a的地方都代表10
var b = a; // 把a的值赋予给了b (b现在也是10了) - 复制
console.log(a, b); // 打印a和b变量里的值

// 答案:
// a - 10
// b - 10
```

2.5 交换2个变量的值

```
// 需求：把a变量的值和b变量的值交换
// 思路：要把a的值赋予到一个中间变量上，再把b的值覆盖给a，然后把中间变量上的值赋予给b
var a = 18;
var b = 31;

var temp; // 声明中间变量temp，先不给值
temp = a; // 把a的值赋予到temp上（temp里装着18）
a = b;    // 把b的值，赋予给a（现在a和b的值都是31了）
b = temp; // 把temp的18，赋予给b（b是18了，a还是31，完成交换）

console.log(a, b);

// 答案：
// a - 31
// b - 18
```

3.2 案例 – 工资计算

效果演示:

Day01/1-代码/3.2_案例_工资计算1.html

127.0.0.1:5500 显示

请输入狗蛋的绩效工资

确定

取消

```
// 需求：老板在输入提示框里，输入狗蛋的绩效工资，然后和固定基本工资相加后，在弹窗中显示狗蛋的最终工资
// 思路：
// (1)：输入提示框(点击确定后) - 可以用变量接受输入提示框返回的数值
// (2)：在JS声明变量，保存固定的基本工资
// (3)：再用alert打印，上面2个变量的和

// 1. 弹窗 - 接受输入内容
var a = prompt("请输入狗蛋的绩效工资");

// 2. 基本工资
var b = 2000;

// 3. 弹窗提示和
alert(a + b);

// 问题：发现a和b拼接了，不是算术+运算
// 原因：页面上获取的值是字符串，字符串和数字相加是拼接
```

```
// "1000" + 2000 ----> "10002000"
console.log(typeof a); // string
console.log(typeof b); // number
```

3.4 案例 - 工资计算2

效果演示:

Day01/1-代码/3.4_案例_工资计算2.html

127.0.0.1:5500 显示

请输入狗蛋的绩效工资

确定

取消

```
// 解决上面遇到的问题
// 3. 修改 - 弹窗提示和
alert(Number(a) + b);
```

作业

作业1 - 弹出钱包余额

/1.html

此网页显示

钱包余额

确定

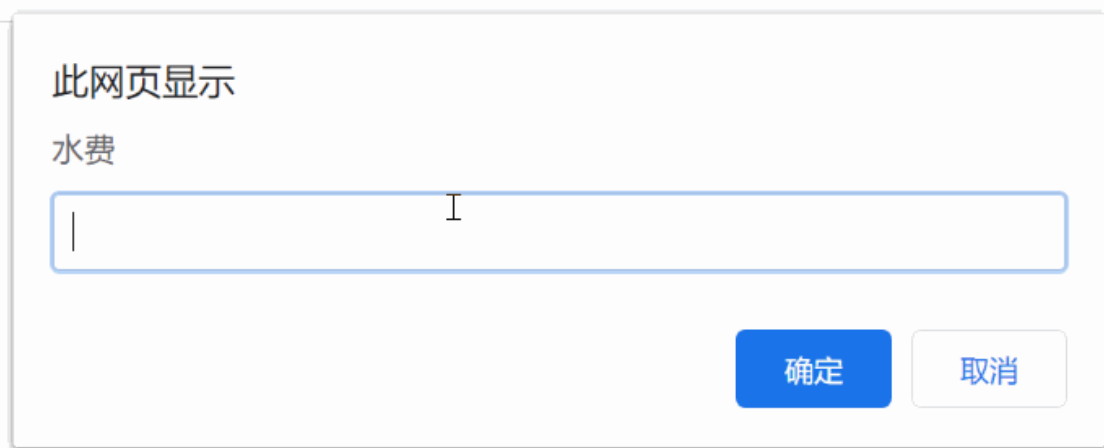
取消

```
// 需求：输入钱包余额和各种费用最后一个弹窗显示钱包余额-各种支出最后的剩余
// 思路：五个弹窗，第一个弹窗输入你钱包里的余额，第二个输入水费，第三个输入电费，第四个输入伙食费，最后
alert返回扣减后，最后钱包里的余额
```

```
// 答案：
// var money = prompt("钱包余额");
// var a = prompt("水费");
// var b = prompt("电费");
// var c = prompt("伙食费");
// alert(money - a - b - c);
```

作业2 - 输入3种费用显示和

/1.html



```
// 需求：请使用一个变量sum，以及赋值运算符+=，实现3个弹窗(水费，电费，伙食费)之和，最后在控制台打印和
即可
```

```
// 答案：
// var sum = 0;
// var a = prompt("水费");
// sum += Number(a);
// var b = prompt("电费");
// sum += Number(b);
// var c = prompt("伙食费");
// sum += Number(c);
// console.log(sum);
```

Day02

课上练习

7.1 for各种练

```
// 1. 练习1: 请打印0-20之间, 每个整数到控制台
// 2. 练习2: 请打印30-100之间, 每个整数到控制台
// 3. 练习3: 请打印1-100之间, 能被3整数的数 - (可以用if哦, 用以前知识点)
// 4. 练习4: 请用for循环, 计算1-10之间每个整数的和
// 5. 练习5: 请打印10-20之间, 所有的偶数的和
```

// 答案:

// 练习1:

```
for (var i = 0; i <= 20; i++) {
    console.log(i);
}
```

// 练习2:

```
for (var i = 30; i <= 100; i++) {
    console.log(i);
}
```

// 练习3:

```
for (var i = 1; i <= 100; i++) {
    if (i % 3 === 0) {
        console.log(i);
    }
}
```

// 练习4:

```
var sum = 0;
for (var i = 1; i <= 10; i++) {
    sum = sum + i; // 每次取出sum的值和当前的i相加
    console.log(sum);
}
```

// 练习5:

```
var sum = 0;
for (var i = 10; i <= 20; i++) {
    if (i % 2 === 0) {
        sum = sum + i;
    }
}
console.log(sum);
```

案例

暂无

作业

作业1 - 分支使用

```
// 需求：定义一个变量表示“考试分数”并给其赋数字分数，在页面中是根据分数高低不同输出如下文字：(*)
// 优秀 (>=90)，良好 (>=80)，中等 (>=70)，及格 (>=60)，不及格
// 代码要求：
// 要求：请分别用if+else / switchcase实现
// 提示：switch/case：因为是===比较，所以对成绩除10，取整，和9，8，7，6判断
```

```
// 答案：
// if + else嵌套
var theGrade = 71;
if (theGrade >= 90) {
    console.log("优秀");
} else if (theGrade >= 80) {
    console.log("良好");
} else if (theGrade >= 70) {
    console.log("中等");
} else if (theGrade >= 60) {
    console.log("及格");
} else {
    console.log("不及格");
}

// switch - case使用
switch (parseInt(theGrade / 10)) {
    case 9:
        console.log("优秀");
        break;
    case 8:
        console.log("良好");
        break;
    case 7:
        console.log("中等");
        break;
    case 6:
        console.log("及格");
        break;
    default:
        console.log("不及格");
}
```

作业2 - 分支练习

```
// 需求：根据一个人的存款的多少决定使用什么交通工具（难度*）
// 如果我有10万元以上，就买小汽车上班
// 或者如果我有5000元以上，我就买摩托上班，
// 或者如果我有300元以上，我就买自行车上班，
// 或者如果我有20元以上，我就坐公交上班，
// 否则，我就只好步行上班（只用if和else实现即可，没有规律不用switch+case）
// 提示：使用一个变量表示其存款数，对该变量进行判断以输出所采用的交通工具
```

```
// 答案:
// if-else使用
var money = 10;
if (money >= 100000) {
    console.log("买小汽车上班");
} else if (money >= 5000) {
    console.log("买摩托上班");
} else if (money >= 300) {
    console.log("买自行车上班");
} else if (money >= 20) {
    console.log("坐公交上班");
} else {
    console.log("走路去上班");
}
```

作业3 - 闰年

// 输出从2000年（含）到3000年（含）间的所有闰年。（难度**）
// 提示：闰年就是能被4整除且不能被100整除的年，或者能被400整除的年。

```
// 答案
for (var i = 2000; i <= 3000; i++) {
    if (i % 4 == 0 && i % 100 !== 0 || i % 400 == 0) {
        console.log(i);
    }
}
```

作业4 - for和if结合使用

//（有难度 - 好好思考 - 尝试拆分问题）难度(***)
// 需求：写一个程序输出1到100这些数字中，遇到数字为3的倍数的时候，输出“三”替代数字，为5的倍数用“五”代替，既是3的倍数又是5的倍数则输出“三五”

```
// 答案
for (var i = 1; i < 100; i++) {
    if (i % 3 == 0 && i % 5 == 0) {
        console.log("三五");
    } else if (i % 3 == 0) {
        console.log("三");
    } else if (i % 5 == 0) {
        console.log("五");
    } else {
        console.log(i);
    }
}
```

作业5 - 99乘法表

```
1 * 1 = 1
2 * 1 = 2 2 * 2 = 4
3 * 1 = 3 3 * 2 = 6 3 * 3 = 9
4 * 1 = 4 4 * 2 = 8 4 * 3 = 12 4 * 4 = 16
5 * 1 = 5 5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25
6 * 1 = 6 6 * 2 = 12 6 * 3 = 18 6 * 4 = 24 6 * 5 = 30 6 * 6 = 36
7 * 1 = 7 7 * 2 = 14 7 * 3 = 21 7 * 4 = 28 7 * 5 = 35 7 * 6 = 42 7 * 7 = 49
8 * 1 = 8 8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64
9 * 1 = 9 9 * 2 = 18 9 * 3 = 27 9 * 4 = 36 9 * 5 = 45 9 * 6 = 54 9 * 7 = 63 9 * 8 = 72 9 * 9 = 81
```

```
// 需求：写99乘法表 - 到页面上输出 难度()
// 使用document.write() 输出到页面上显示 配合<br/>换行，差不多就ok，无需对得特别齐
// 并能讲出代码的执行顺序(可辅助断点调试使用)
```

```
for (var i = 1; i <= 9; i++) {
    for (var j = 1; j <= i; j++) {
        document.write((i + "*" + j) + "=" + i * j);
    }
    document.write("<br/>");
}
```

额外练习

```
// 6. 使用三元运算符，返回3个数里的最大值(难度 ***)
var a = 20;
var b = 5;
var c = 10;
var max = a > b ? (a > c ? a : c) : (b > c ? b : c);
console.log(max);
```

```
// 7. 小李准备在操场跑10圈 - 每次跑一圈 计数 + 1，当小李跑到6圈时，不想跑了，请用代码实现这个过程(难度 *)
```

```
for (var i = 1; i < 11; i++) {
    if (i === 6) {
        break;
    }
    console.log(i);
}
```

```
// 8. 求出1 - 10之间 除了5的乘积(考虑使用continue)(难度 **)
```

```
var c = 1;
for (var i = 1; i < 11; i++) {
    if (i === 5) {
        continue;
    }
    c = c * i;
}
```



```

    }
    c = c * i;
}
console.log(c);

// 9. 100米高度落下，每次反弹一半，第10次经过多少米，第10次反弹多高？(难度 ****)
var now = 100;
var sum = 0;
for (var i = 1; i < 11; i++) {
    // 落下过程
    sum = sum + now;
    // 反弹
    now = now / 2;
    // 把反弹的距离加上
    sum = sum + now;
}
console.log(sum, now);

// 10. 打印水仙花数100 - 999 每位立方和等于本身的数字(难度 ***)
for (var i = 100; i < 1000; i++) {
    var bai = Math.floor(i / 100);
    var shi = Math.floor(i % 100 / 10);
    var ge = Math.floor(i % 10);
    if (Math.pow(bai, 3) + Math.pow(shi, 3) + Math.pow(ge, 3) === i) {
        console.log(i);
    }
}

// 11. 请从 100 输出到 -100 注意从大往小输出(难度 ***)
for (var i = 100; i >= -100; i--) {
    console.log(i);
}

// 12. 请输出 1 - 100之间 既能被5整除也能被20整除的数，打印这些值(难度 **)
for (var i = 1; i < 101; i++) {
    if (i % 5 === 0 && i % 20 === 0) {
        console.log(i);
    }
}

// 13. 请计算上一题，所有数的和(难度 ***)
var sum = 0;
for (var i = 1; i < 101; i++) {
    if (i % 5 === 0 && i % 20 === 0) {
        sum += i;
    }
}
console.log(sum);

```

Day03

课上练习

10.2 for + if和else + 数组练习

```
// 练习1: 使用for循环 - 求出数组元素的和 [5, 8, 9, 2, 1, 5]
// 练习2: 使用for循环 - 求出数组里大于5的和 [4, 9, 5, 20, 3, 11]
// 练习3: 使用for循环 - 求出班级里同学们平均年龄[15, 19, 21, 33, 18, 24]
// 练习4: 打印数组里所有的奇数, [5, 2, 8, 10, 3, 7]
// 练习5: 打印数组里所有能被5整除的奇数, [10, 15, 80, 75]
// 练习6: 计算[2, 6, 18, 15, 40] 中能被3整除的偶数的和
// 练习7: 计算[2, 6, 18, 15, 40] 中能被3整除的偶数的个数
```

```
// 答案:
```

```
// 练习1:
```

```
var arr = [5, 8, 9, 2, 1, 5];
var sum = 0;
for (var i = 0; i < arr.length; i++) {
    sum = sum + arr[i]; // 不再加i了, 而是通过i从数组里换出的元素
}
console.log(sum);
```

```
// 练习2:
```

```
var arr = [4, 9, 5, 20, 3, 11];
var sum = 0;
for (var i = 0; i < arr.length; i++) {
    if (arr[i] > 5) {
        sum = sum + arr[i];
    }
}
console.log(sum);
```

```
// 练习3
```

```
// 先求和
```

```
// 再for结束后, 把和除以数组的长度
```

```
var arr = [15, 19, 21, 33, 18, 24];
var sum = 0;
for (var i = 0; i < arr.length; i++) {
    sum = sum + arr[i];
}
console.log(sum / arr.length);
```

```
// 练习4
```

```
var arr = [5, 2, 8, 10, 3, 7];
for (var i = 0; i < arr.length; i++) {
    if (arr[i] % 2 !== 0) {
        console.log(arr[i]);
    }
}
```

```
// 练习5
```

```
var arr = [10, 15, 80, 75];
for (var i = 0; i < arr.length; i++) {
    if (arr[i] % 5 == 0 && arr[i] % 2 != 0) {
        console.log(arr[i]);
    }
}

// 练习6
var arr = [2, 6, 18, 15, 40];
var sum = 0;
for (var i = 0; i < arr.length; i++) {
    if (arr[i] % 3 == 0 && arr[i] % 2 == 0) {
        sum += arr[i];
    }
}
console.log(sum);

// 练习7
var arr = [2, 6, 18, 15, 40];
var len = 0;
for (var i = 0; i < arr.length; i++) {
    if (arr[i] % 3 == 0 && arr[i] % 2 == 0) {
        len += 1;
    }
}
console.log(len);
```

案例

11.0 小娜v1.0 - 介绍引导

Day03/1_代码/11.0_案例_小娜_介绍引导.html

127.0.0.1:5500 显示

你好我是小娜V1，请选择功能：

输入q:退出;

输入1:求和;

输入2:系统时间;

输入3:随机笑话;

确定

取消

// 需求：只要用户输入的不是相关选项，则一直存在弹出输入提示框，等待用户的输入

// 思路：

// (1): while(true) - 死循环

```
// (2): 弹出输入框
```

```
while (true) {  
    prompt("你好我是小娜V1, 请选择功能: \n 输入q:退出; \n 输入1:求和; \n 输入2:系统时间; \n 输入3:随机笑话;");  
}
```

```
// 代码执行顺序:
```

```
// while(true) -> 成立 -> 进入 -> 弹窗
```

```
// 点击确定/取消 -> 代码继续执行
```

```
// while循环又是true -> 所以继续执行一次prompt弹窗
```

```
// ..往复下去死循环
```

11.1 小娜v1.0 - 退出功能

Day03/1_代码/11.4_案例_小娜_随机笑话.html

127.0.0.1:5500 显示

你好我是小娜V1, 请选择功能:

输入q:退出;

输入1:求和;

输入2:系统时间;

输入3:随机笑话;

确定

取消

```
// 需求: 当用户输入q字符的时候, alert弹出一个告别文字, 然后不再弹出弹窗
```

```
// 思路:
```

```
// (1): 在上一个基础上, 接受prompt() 返回的值
```

```
// (2): 在下面根据用户输入的值, 进行if判断 - 执行相应代码
```

```
// (3): 退出循环应该使用什么关键字呀?
```

```
// 死循环
```

```
while (true) {  
    var num = prompt("你好我是小娜V1, 请选择功能: \n 输入q:退出; \n 输入1:求和; \n 输入2:系统时间; \n 输入3:随机笑话;");  
    if (num == "q") {  
        alert("我退出咯, 你不爱我了!");  
        break;  
    }  
}
```

11.2 小娜v1.0 - 求和功能

Day03/1_代码/11.4_案例_小娜_随机笑话.html

127.0.0.1:5500 显示

你好我是小娜V1, 请选择功能:

输入q:退出;

输入1:求和;

输入2:系统时间;

输入3:随机笑话;

I

确定

取消

```
// 需求: 根据用户prompt输入返回值, 判断, 然后在相应if里, 执行对应的代码
// 思路:
// (1): 在上一个基础上, 接受prompt() 返回的值
// (2): 在下面根据用户输入的值, 进行if判断 - 执行相应代码
// (3): 这时, 我们还需要一个输入框, 让用户输入等待计算的多个数字(用什么隔开, 可以自行添加规则, 这里选择逗号)
// (4): 然后用alert弹出计算的和结果, 代码执行完毕, 还会进行while(true)的死循环, 还会有弹窗出现

// 死循环
while (true) {
    var num = prompt("你好我是小娜V1, 请选择功能: \n 输入q:退出; \n 输入1:求和; \n 输入2:系统时间; \n 输入3:随机笑话;");
    if (num == "q") {
        alert("我退出咯, 你不爱我了!");
        break;
    }
    // 输入1 - 求和
    if (num == 1) {
        var result = prompt("请输入数字, 中间用逗号隔开");
        var arr = result.split(",");
        var sum = 0;
        for (var i = 0; i < arr.length; i++) {
            sum = sum + parseInt(arr[i]);
        }
        alert(sum);
    }
}

// 详细解释:
// 当if为1的里面代码执行完毕, 注意最外面还套着一个while(true)
// 所以while还会继续执行, 等待用户输入prompt(), 如果输入错误/点击确定/点击取消
// 代码往下走, 因为num值匹配不了2个if所以都进不去
// 这时, 又会从新执行一次while循环
```

11.3 小娜v1.0 - 显示系统时间

考查知识点:

- 获取本机系统时间 `new Date()`
- 获取其中数字 调用各种方法
- 会判断然后前置补0字符, 改变变量

Day03/1_代码/11.4_案例_小娜_随机笑话.html

127.0.0.1:5500 显示

你好我是小娜V1, 请选择功能:

输入q:退出;

输入1:求和;

输入2:系统时间;

输入3:随机笑话;

确定

取消

```
// 需求: 根据用户prompt输入返回值, 判断, 然后在相应if里, 执行对应的代码
// 思路:
// (1): 在上一个基础上, 接受prompt() 返回的值
// (2): 在下面根据用户输入的值, 进行if判断 - 执行相应代码
// (3): 获取系统时间 new Date()
// (4): 拿到时间上的每一部分 (需要掌握对应方法使用)
```

```
// 死循环
```

```
while (true) {
    var num = prompt("你好我是小娜V1, 请选择功能: \n 输入q:退出; \n 输入1:求和; \n 输入2:系统时间; \n 输入3:随机笑话;");
    if (num == "q") {
        alert("我退出咯, 你不爱我了!");
        break;
    }
    // 输入1 - 求和
    if (num == 1) {
        var result = prompt("请输入数字, 中间用逗号隔开");
        var arr = result.split(",");
        var sum = 0;
        for (var i = 0; i < arr.length; i++) {
            sum = sum + parseInt(arr[i]);
        }

        alert(sum);
    }
}
```

```

}
// 输入2 - 输出系统时间
if (num == 2) {
    var now = new Date(); // 获取本机系统时间
    var year = now.getFullYear(); // 返回年的4位
    var month = now.getMonth() + 1; // 获取月份从0开始，所以+1
    var day = now.getDate(); // 返回几号
    var hours = now.getHours();
    var minutes = now.getMinutes();
    var seconds = now.getSeconds();

    // 前补0
    if (month < 10) {
        month = "0" + month; // 假如month为4，那么这行代码执行后month值就为"04"
    }
    day = day < 10 ? '0' + day : day;
    hours = hours < 10 ? '0' + hours : hours;
    minutes = minutes < 10 ? '0' + minutes : minutes;
    seconds = seconds < 10 ? '0' + seconds : seconds;

    // alert(year + "/" + month + "/" + day + " " + hours + ":" + minutes + ":" + seconds);
    alert(`${year}/${month}/${day} ${hours}:${minutes}:${seconds}`);
}
}

```

11.4 小娜v1.0 - 随机笑话

Day03/1_代码/11.4_案例_小娜_随机笑话.html

127.0.0.1:5500 显示

你好我是小娜V1，请选择功能：

输入q:退出;

输入1:求和;

输入2:系统时间;

输入3:随机笑话;

确定

取消

// 需求：用户输入3，随机显示一条笑话 ["'生'的反义词是什么，一般人都会想到'死'，出货就和你想的不一样，人家觉得是'熟'，" "你好烦人，废话，我不是凡人，我难道是神仙?"， "从前有个北极熊，无聊拔毛玩，拔着拔着就冻死了"， "主人，还是你给我讲个笑话吧，虽然我也听不懂"]

// 思路：

// (1)：先在while里，加入3的判断 - 在if大括号里编写相关代码

// (2)：先准备好，一些笑话在数组里

// (3)：学会随机数公式 - 背下来

// (4)：随机产生一个小标数，然后换出笑话的字符串 - 弹出显示

```

// 死循环
while (true) {
    var num = prompt("你好我是小娜v1, 请选择功能: \n 输入q:退出; \n 输入1:求和; \n 输入2:系统时间; \n 输入3:随机笑话;");
    if (num == "q") {
        alert("我退出咯, 你不爱我了!");
        break;
    }
    // 输入1 - 求和
    if (num == 1) {
        var result = prompt("请输入数字, 中间用逗号隔开");
        var arr = result.split(",");
        var sum = 0;
        for (var i = 0; i < arr.length; i++) {
            sum = sum + parseInt(arr[i]);
        }
        alert(sum);
    }
    // 输入2 - 输出系统时间
    if (num == 2) {
        var now = new Date("2019/02/03 09:09:04"); // 注意格式:手动修改代码中获取的时间
        var year = now.getFullYear(); // 返回年的4位
        var month = now.getMonth() + 1; // 获取月份从0开始, 所以+1
        var day = now.getDate(); // 返回几号
        var hours = now.getHours();
        var minutes = now.getMinutes();
        var seconds = now.getSeconds();

        // 前补0
        if (month < 10) {
            month = "0" + month; // 假如month为4, 那么这行代码执行后month值就为"04"
        }
        day = day < 10 ? '0' + day : day;
        hours = hours < 10 ? '0' + hours : hours;
        minutes = minutes < 10 ? '0' + minutes : minutes;
        seconds = seconds < 10 ? '0' + seconds : seconds;

        // alert(year + "/" + month + "/" + day + " " + hours + ":" + minutes + ":" + seconds);
        alert(`${year}/${month}/${day} ${hours}:${minutes}:${seconds}`);
    }
    // 输入3 - 随机显示笑话
    if (num == 3) {
        var arr = ["'生'的反义词是什么, 一般人都会想到'死', 出货就和你想的不一樣, 人家觉得是'熟'", "你  
好烦人, 废话, 我不是凡人, 我难道是神仙?", "从前有个北极熊, 无聊拔毛玩, 拔着拔着就冻死了", "主人, 还是你  
给我讲个笑话吧, 虽然我也听不懂"];
        var randIndex = Math.floor(Math.random() * (arr.length - 1 - 0) + 0);
        alert(arr[randIndex]);
    }
}

```

作业

作业 - for+if+数组 大量练习

```
// 作业1: 统计班级学生年龄: [18, 20, 21, 28, 31, 17, 24], 计算总年龄 和 平均年龄
// 作业2: 给一个不重复的数字数组[18, 20, 21, 28, 31, 17, 24], 求最大值和最大值的在数组中的位置
// 作业3: 给一个不重复的数字数组[18, 20, 21, 28, 31, 17, 24], 求最大值和最大值的在数组中的位置 和 最小值和最小值的在数组中的位置
// 作业4: 给一个字符串数组["张三", "李四", "王五"], 将数组中的每一个数据拼接在一起, 每个数据之间用 | 符合拼接。 如: '张三|李四|王五'
// 作业5: 给一个数字数组[1, 0, 2, 0, 0, 0, 5], 该数组中有很多数字0, 将不为0的数据存入到一个新的数组中
// 作业6: 给一个数组[1, 0, 2, 0, 0, 0, 5], 将数组中的数据反转, 存放到一个新的数组中
// 作业7: 假设我有一千万, 一天花一半, 问我能花多少天(少于1元就不能花了)
// 作业8: 点名: 每次刷新网页运行, 在控制台 随机输出一位同学的名字 ["老赵", "老李", "小传", "小黑"]
// 作业9: 以这个格式 - 打印今天日期: xxxx年xx月xx日 - xx时xx分xx秒
```

// 答案:

```
// 作业1: 统计班级学生年龄: [18, 20, 21, 28, 31, 17, 24], 计算总年龄 和 平均年龄
```

```
var arr = [18, 20, 21, 28, 31, 17, 24];
var allAge = 0;
for (var i = 0; i < arr.length; i++) {
    allAge += arr[i];
}
console.log(allAge);
console.log(allAge / arr.length);
```

```
// 作业2: 给一个不重复的数字数组[18, 20, 21, 28, 31, 17, 24], 求最大值和最大值的在数组中的位置
```

```
var arr = [18, 20, 21, 28, 31, 17, 24];
var maxIndex = -1;
var max = 0;
for (var i = 0; i < arr.length; i++) {
    if (arr[i] > max) {
        max = arr[i];
        maxIndex = i;
    }
}
console.log(maxIndex, max);
```

```
// 作业3: 给一个不重复的数字数组[18, 20, 21, 28, 31, 17, 24], 求最大值和最大值的在数组中的位置 和 最小值和最小值的在数组中的位置
```

```
var arr = [18, 20, 21, 28, 31, 17, 24];
var maxIndex = -1;
var max = 0;
var min = arr[0];
var minIndex = -1;
for (var i = 0; i < arr.length; i++) {
    if (arr[i] > max) {
        max = arr[i];
        maxIndex = i;
    }
    if (arr[i] < min) {
        min = arr[i];
        minIndex = i;
    }
}
```

```

}
console.log(maxIndex, max);
console.log(minIndex, min);

// 作业4: 给一个字符串数组["张三", "李四", "王五"], 将数组中的每一个数据拼接在一起, 每个数据之间用 | 符合拼接。 如: '张三|李四|王五'
var arr = ["张三", "李四", "王五"];
var str = "";
for (var i = 0; i < arr.length; i++) {
    if (i !== arr.length - 1){
        str += arr[i] + "|";
    }
}
console.log(str);

// 作业5: 给一个数字数组[1, 0, 2, 0, 0, 0, 5], 该数组中有很多数字0, 将不为0的数据存入到一个新的数组中
var arr = [1, 0, 2, 0, 0, 0, 5];
var brr = [];
for (var i = 0; i < arr.length; i++) {
    if (arr[i] !== 0) {
        brr.push(arr[i]);
    }
}
console.log(brr);

// 作业6: 给一个数组[1, 0, 2, 0, 0, 0, 5], 将数组中的数据反转, 存放到一个新的数组中
var arr = [1, 0, 2, 0, 0, 0, 5];
var brr = [];
for (var i = arr.length - 1; i >= 0; i--) {
    brr.push(arr[i]);
}
console.log(brr);

// 作业7: 假设我有一千万, 一天花一半, 问我能花多少天(少于1元就不能花了)
var money = 10000000;
var day = 0;
for (; money >= 1; ){
    money -= money / 2;
    day++;
}
console.log(day);

// 作业8: 点名: 每次刷新网页运行, 在控制台 随机输出一位同学的名字 ["老赵", "老李", "小传", "小黑"]
var arr = ["老赵", "老李", "小传", "小黑"];
console.log(arr[Math.floor(Math.random() * arr.length)]);

// 作业9: 以这个格式 - 打印今天日期: xxxx年xx月xx日 - xx时xx分xx秒
var now = new Date();
var year = now.getFullYear(); // 返回年的4位
var month = now.getMonth() + 1; // 获取月份从0开始, 所以+1
var day = now.getDate(); // 返回几号

var hours = now.getHours();

```

```
var minutes = now.getMinutes();
var seconds = now.getSeconds();

// 前补0
month = month < 10 ? '0' + month : month;
day = day < 10 ? '0' + day : day;
hours = hours < 10 ? '0' + hours : hours;
minutes = minutes < 10 ? '0' + minutes : minutes;
seconds = seconds < 10 ? '0' + seconds : seconds;

alert(`${year}年${month}月${day}日 ${hours}时${minutes}分${seconds}秒`);
```

Day04

课上练习

12.6 函数封装练习

```
// 练习1: 尝试封装函数 - 实现数组里求最大值的函数
// 练习2: 尝试封装函数 - 求数组里和的平均值的函数
// 练习3: 尝试封装函数 - 传入数组(里面都是人名) - 每次调用返回一个随机人名

// 答案:
// 练习1: 尝试封装函数 - 实现数组里求最大值的函数
function getMax(arr){
    var max = 0;
    for (var i = 0; i < arr.length; i++) {
        max = arr[i] > max ? arr[i] : max;
    }
    return max;
}
console.log(getMax([5, 3, 10, 2, 1]));
// 练习2: 尝试封装函数 - 求数组里和的平均值的函数
function getAvg(arr){
    var sum = 0;
    for (var i = 0; i < arr.length; i++) {
        sum += arr[i];
    }
    return sum / arr.length;
}
console.log(getAvg([5, 3, 10, 2, 1]));

// 练习3: 尝试封装函数 - 传入数组(里面都是人名) - 每次调用返回一个随机人名
function getRandName(arr){
    return arr[Math.floor(Math.random() * arr.length)];
}
console.log(getRandName(["张三", "李四", "王五"]));

console.log(getRandName(["小黑", "小传", "小马"]));
```

12.8 作用域链

// 需求: 请回答下面代码打印的结果

```
var a = 100;
function one(){
  console.log(a);
  function two(){
    var a = 50;
    console.log(a);
    function three(){
      console.log(a);
      function four(){
        console.log(a);
        function five(){
          var a = 20;
          console.log(a);
        }
        five();
      }
      four();
    }
    three();
  }
  two();
}
one();
```

// 答案:

```
var a = 100;
function one(){
  console.log(a); // 100
  function two(){
    var a = 50;
    console.log(a); // 50
    function three(){
      console.log(a); // 50
      function four(){
        console.log(a); // 50
        function five(){
          var a = 20;
          console.log(a); // 20
        }
        five();
      }
      four();
    }
    three();
  }
  two();
}
```

```
one();
```

13.2 变量提升+作用域链

- 考查知识点
 - 变量提升
 - 作用域

```
// 需求：回答下面的结果
var num = 10;
fun();
console.log(num);

function fun() {
  console.log(num);
  var num = 20;
  console.log(num);
}

// 答案，以及讲解
// 模拟预解析后的代码
// (1)：从头带尾看代码 - 先确定作用域
// (2)：预解析 - 提升所有变量
// (3)：在从头到尾 - 运行每行代码
var num = 10;
fun();
console.log(num); // 10

function fun() {
  console.log(num); // undefined
  var num = 20;
  console.log(num); // 20
}
```

13.3 变量提升+作用域链

```
// 重点考查作用域使用，以及变量提升
var num = 10;
fun();
console.log(num);

function fun() {
  console.log(num);
  num = 20;
  console.log(num);
}

// 答案 - 执行过程
var num = 10;
fun();
```

```

console.log(num); // 20

function fun() {
    console.log(num); // 10
    num = 20; // 这没有var，使用的是全局变量(修改的也是全局变量，这里所有num都是全局变量num的值)
    console.log(num); // 20
}

```

案例

效果gif与昨天Day03的小娜v1.0相同 - 这里就不再演示

只是代码变成了函数封装

14.0 小娜v2.0 - 封装函数 - 求和功能

```

/**
 * 此方法用于求和
 * @param {*} result - 逗号分隔的多个数字
 * @returns 累计的和
 */
function getSum(result){
    var arr = result.split(",");
    var sum = 0;
    for (var i = 0; i < arr.length; i++) {
        sum = sum + parseInt(arr[i]);
    }
    return sum;
}

// 小娜的代码
if (num == 1) {
    var result = prompt("请输入数字，中间用逗号隔开");
    alert(getSum(result));
}

```

14.1 小娜v2.0 - 封装函数 - 系统时间

```

/**
 * 打印当前系统时间
 * @returns {String} - 系统日期字符串 格式:"xxxx年xx月xx日 时:分:秒"
 */
function getNowDate() {
    var now = new Date(); // 注意格式:手动修改代码中获取的时间
    var year = now.getFullYear(); // 返回年的4位
    var month = now.getMonth() + 1; // 获取月份从0开始，所以+1
    var day = now.getDate(); // 返回几号
    var hours = now.getHours();
    var minutes = now.getMinutes();
    var seconds = now.getSeconds();
}

```

```

// 前补0
if (month < 10) {
    month = "0" + month; // 假如month为4, 那么这行代码执行后month值就为"04"
}
day = day < 10 ? '0' + day : day;
hours = hours < 10 ? '0' + hours : hours;
minutes = minutes < 10 ? '0' + minutes : minutes;
seconds = seconds < 10 ? '0' + seconds : seconds;

return `${year}/${month}/${day} ${hours}:${minutes}:${seconds}`;
}

// 小娜处代码换成:
if (num == 2) {
    alert(getNowDate());
}

```

14.2 小娜v2.0 - 封装函数 - 随机笑话

```

/**
 *
 * @param {Array[String]} arr - 数组里放入的字符串
 * @returns {String} - 随机返回一个字符串
 */
function randStrFromArr(arr) {
    return arr[Math.floor(Math.random() * arr.length)];
}

// 小娜代码修改
if (num == 3) {
    var arr = ["'生'的反义词是什么, 一般人都会想到'死', 出货就和你想的不一樣, 人家觉得是'熟'", "你好烦人, 废话, 我不是凡人, 我难道是神仙?", "从前有个北极熊, 无聊拔毛玩, 拔着拔着就冻死了", "主人, 还是你给我讲个笑话吧, 虽然我也听不懂"];
    alert(randStrFromArr(arr));
}

```

作业

```

// 作业1. 限时代码 - 数组声明提前准备好 - 请在1分钟之内 - 完成冒泡排序代码(不用声明, 不用打印只写核心带-i和-1的 (边写-边思考-20遍以后你就明白了冒泡)

// 作业2. 封装函数- 实现调用后 - 在页面生成9*9乘法表 (页面效果跟之前写的一样)
function createNine() {
    for (var i = 1; i <= 9; i++) {
        for (var j = 1; j <= i; j++) {
            document.write((i + "*" + j) + "=" + i * j);
        }
        document.write("<br/>");
    }
}

```

```
createNine();
createNine(); // 调用一次 就来个9*9乘法表
```

Day05

课上练习

暂无

案例

小娜V3.0, 在v2.0基础上, 把函数封装到对象里, 用对象来调用函数, 效果与之前一样, 还是更改代码即可

17.0 小娜v3.0 - 对象化 - 求和功能

```
var toolObj = {};  
toolObj.getSum = function (result) {  
    var arr = result.split(",");  
    var sum = 0;  
    for (var i = 0; i < arr.length; i++) {  
        sum = sum + parseInt(arr[i]);  
    }  
    return sum;  
}  
  
// 小娜处修改  
if (num == 1) {  
    alert(toolObj.getSum(prompt("请输入数字，中间用逗号隔开")));  
}
```

17.1 小娜v3.0 - 对象化 - 系统时间功能

```
toolObj.getNowDate = function () {  
    var now = new Date(); // 注意格式:手动修改代码中获取的时间  
    var year = now.getFullYear(); // 返回年的4位  
    var month = now.getMonth() + 1; // 获取月份从0开始, 所以+1  
    var day = now.getDate(); // 返回几号  
    var hours = now.getHours();  
    var minutes = now.getMinutes();  
    var seconds = now.getSeconds();  
  
    // 前补0  
    if (month < 10) {  
        month = "0" + month; // 假如month为4, 那么这行代码执行后month值就为"04"  
    }  
    day = day < 10 ? '0' + day : day;  
    hours = hours < 10 ? '0' + hours : hours;  
    minutes = minutes < 10 ? '0' + minutes : minutes;  
    seconds = seconds < 10 ? '0' + seconds : seconds;
```



```

        return `${year}/${month}/${day} ${hours}:${minutes}:${seconds}`;
    }

    // 小娜处修改
    if (num == 2) {
        alert(toolObj.getNowDate());
    }
}

```

17.2 小娜v3.0 - 对象化 - 随机笑话

```

// 1. 先把笑话数组绑定到对象的身上 - 封装在工具里
var toolObj = {
    jokeArr: [ "'生'的反义词是什么，一般人都会想到'死'，出货就和你想的不一样，人家觉得是'熟'，"，"你好烦人，废话，我不是凡人，我难道是神仙？"，"从前有个北极熊，无聊拔毛玩，拔着拔着就冻死了"，"主人，还是你给我讲个笑话吧，虽然我也听不懂" ]
};

// 2. 修改封装的返回笑话的函数
/**
 *
 * @returns {String} - 随机返回一个字符串
 */
toolObj.randStrFromArr = function () {
    return toolObj.jokeArr[Math.floor(Math.random() * toolObj.jokeArr.length)];
}

// 小娜处修改
// 输入3 - 随机显示笑话
if (num == 3) {
    alert(toolObj.randStrFromArr());
}

```

17.3 小娜v3.0 - 对象化 - 新功能 - 求n-m之间的随机整数

127.0.0.1:5500 显示

你好我是小娜V1, 请选择功能:

输入q:退出;

输入1:求和;

输入2:系统时间;

输入3:随机笑话;

输入4:取随机数

输入5: 录入新笑话

确定

取消

比如, 今天66个同学, 谁去取餐, 大家编号 - 随机一个

```
// 对象里新增方法功能
/**
 *
 * @param {String} str - 数字1,数字2
 * @returns {Number} - 返回上面2个数字范围的随机整数
 */
toolObj.getRandWith = function(str) {
    var arr = str.split(",");
    var n = arr[0];
    var m = arr[1];
    if (n > m) { // 如果第一个数是大的, 交换
        var temp = n;
        n = m;
        m = temp;
    }
    // 因为公式里n位置必须是小值
    return Math.floor(Math.random() * (m - n + 1) + n);
}

// 小娜提示面板里, 新增数字4
// 输入4 - 取指定返回的随机数
if (num == 4) {
    alert(toolObj.getRandWith(prompt("请以1,5这种格式, 输入随机数范围")))
}
```

17.4 小娜 - 对象化 - 添加笑话功能

127.0.0.1:5500 显示

你好我是小娜V1, 请选择功能:

输入q:退出;

输入1:求和;

输入2:系统时间;

输入3:随机笑话;

输入4:取随机数

输入5: 录入新笑话

I

确定

取消

前提知识点: 每次刷新网页, 所有代码都会重新执行.

在网页运行过程中, 所有的操作 - 在刷新页面后 - 都回归最初状态 - 变量(临时保存) - 以后会学数据持久化

// 需求: 小娜打开运行过程中 - 可以根据指示 - 动态批量录入新的笑话 - 然后在随机笑话时, 可以显示新录入的和原来的笑话

// 注意: 新录入的临时保存在数组变量中 - 刷新网页还是只有代码里写死的笑话

// 思路:

// 在对象上 - 添加方法 - 接受新笑话 - 指定格式哦

// 向对象上现有的笑话数组中, 把新录入的笑话一个个追加push方法进去

// 新增方法

```
toolObj.addJoke = function(str) {  
    var arr = str.split("|");  
    for (var i = 0; i < arr.length; i++) {  
        toolObj.jokeArr.push(arr[i]);  
    }  
}
```

// 输入5 - 录入新笑话-注意格式

```
if (num == 5) {  
    toolObj.addJoke(prompt("多个笑话用|隔开, 例如: 笑话1|笑话2|笑话3"));  
}
```

作业

- 考试知识点:
 - 数组和对象结构的认知
 - 遍历数组的方式
 - 判断的使用

```
// 自己定义数组 - 里面有3个对象 (要求每个对象里 有姓名, 年龄, 性别),
// 需求1: 遍历数组里每个对象, 判断出哪个人最大 - 打印它的名字
// 需求2: 遍历数组里每个对象, 为每个人增加5岁的年龄, 最后打印整个数组在控制台查看
```

```
// 需求: 声明数组, 里面放入3个对象 (姓名, 年龄, 性别)
// (1): 求年龄最大的人的 名字
// (2): 给每个人年龄+5岁
// 重要: 把数组里的每个对象(当成一个整体去看)
```

```
var arr = [
  {
    name: "小李",
    age: 18,
    sex: "女"
  },
  {
    name: "老王",
    age: 39,
    sex: "男"
  },
  {
    name: "小爱",
    age: 15,
    sex: "女"
  }
];
```

```
// 1.0 求最大值
```

```
var max = 0;
var maxName = null; // 保存最大年龄人的名字
```

```
// 1.1 生成索引
```

```
for (var i = 0; i < arr.length; i++) {
  // 1.2 每次取出的是对象 - 用对象的age属性值 - 和max比较
  if (arr[i].age > max) {
    // 1.3 保存最大的年龄
    max = arr[i].age;
    // 1.4 同时保存它的名字
    maxName = arr[i].name;
  }
}
```

```
// 1.5 for跑完以后, 都比较完毕, 会保存最大的年龄和名字
console.log(max, maxName);
```

```
// 方式2: (用的最多的方法)
```

```
// 还可以保存最大年龄哪个对象的下标 / 把这个对象保存起来
```

```
var max = 0;
var maxInd = null; // 保存最大年龄人的 在数组里的下标
```

```
// 1.1 生成索引
```

```
for (var i = 0; i < arr.length; i++) {
  // 1.2 每次取出的是对象 - 用对象的age属性值 - 和max比较
  if (arr[i].age > max) {
    // 1.3 保存最大的年龄
    max = arr[i].age;
    // 1.4 保存的是下标
    maxInd = i;
  }
}
```

```

    }
}
// 1.5 for跑完以后，都比较完毕，用下角标再去数组里取出最大的那个对象，再访问名字
console.log(max, arr[maxInd].name);

// 需求2: 每个人的年龄+5
// 2.1 把每个对象从 数组里遍历出来
for (var i = 0; i < arr.length; i++) {
    // 2.2 取出对象 arr[i]
    // 2.3 取出当前对象的年龄arr[i].age, 基础上+5, 再赋予回去
    arr[i].age = arr[i].age + 5;
}
console.log(arr);
// 思考: 为什么arr[i] 就能影响原数组里的对象呢?
// 相当于 var arr[i] = {name:"", age: 0, sex: ""}
// var obj = {}; obj.age = 100;
// 这里边 arr[i]接受到的就是数组里对象, 所以只能操作它

// 基础类型 和 复杂类型的 引用关系 的区别:
// 1. 基础类型 - 单纯的值复制 - 再无任何瓜葛
var arr = [10, 20, 50];
for (var i = 0; i < arr.length; i++) {
    var t = arr[i]; // 只是单纯的值的复制 (t和原来的再无任何瓜葛)
    t = 100;
}
console.log(arr); // [10, 20, 50]

// 参照
var arr = [10, 20, 50];
for (var i = 0; i < arr.length; i++) {
    arr[i] = 100; // 没有复制的动作, 是直接操作arr数组
}
console.log(arr); // [100, 100, 100]
// 2. 引用类型 -
var arr = [{name: "老李"}, {name: "老刘"}];
for (var i = 0; i < arr.length; i++) {
    var t = arr[i]; // 引用类型的复制是地址的复制 - t和arr[i]指向的是同一个堆内存地址 (互相影响)
    t.name = "无名氏";
}
console.log(arr); // [{name: "无名氏"}, {name: "无名氏"}];

// 总结: 先找= 赋值的过程, 然后在看类型, 如果是数组/对象, 注意了互相影响

```

Day06

课上练习

暂无

案例

21.1 案例 - 距离高考还有多少天

距离高考还有多少天

// 需求: 计算今天 - 距离下次高考(6月7日) - 还差多少天

// 思路: 先计算毫秒差值 - 然后把毫秒转成天

```
var nowDate = new Date();
```

```
var targetDate = new Date("2021/06/07 09:00:00");
```

// 1970 __A__ 现在

// 1970 _____B_____ 下一次高考

// 要用B - A求出来 现在距离下一次高考的差的毫秒值

```
var seconds = targetDate.getTime() - nowDate.getTime();
```

```
console.log(seconds / 1000 / 60 / 60 / 24);
```

作业

作业1 - 计算出你在世界上活了多少天

// 需求: 在控制台输出你活了多少天

// 提示:

// (1): 可以获取时间对象- 传入你出生的年月日

// (2): 可以获取当前系统时间对象 - 今天的年月日

// (3): 获取2个时间点到1970年的毫秒值, 做减法, 再转换成天就可以了

```
var birthDate = new Date("1981-10-14");
```

```
var nowDate = new Date();
```

```
var chaTime = nowDate.getTime() - birthDate.getTime();
```

```
console.log(chaTime / 1000 / 60 / 60 / 24);
```